Succinct Graph Structures and Applications **Spring 2020**

Exercise 2: May 28

Lecturer: Merav Parter

Additive Spanners

Exercise 1. Given an unweighted n-vertex graph $G = (V, E)$, a 6-additive spanner $H \subseteq G$ is a subgraph satisfying that

$$\text{dist}(u, v, H) \leq \text{dist}(u, v, G) + 6, \forall u, v \in V .$$

We saw in class, the construction of +2 and +4 additive spanners. In this exercise, we will construct, in a step by step manner, a 6-additive spanner H with $\widetilde{O}(n^{4/3})$ edges.

(a) The first algorithm defines a degree threshold Δ_1, and adds all edges incident to vertices with degree at most Δ_1 to H. How large can Δ_1 be (i.e., so that the edge bound of $\widetilde{O}(n^{4/3})$ is kept)?

(b) Next, the algorithm takes care of all shortest paths $\pi(u, v)$ that have at least *one* high-degree vertex with degree at least Δ_2. To do that, it samples each vertex $v \in V$ into a set Q independently with probability $C \cdot \log n/\Delta_2$ for some large constant C. A BFS tree rooted at each vertex $q \in Q$ is added to H. How small can Δ_2 be?

(c) Finally, it remains to take care of paths that have no vertex with degree at least Δ_2. On each path $\pi(u, v)$, observe that all edges incident to low-degree vertices (vertices with degree at most Δ_1) are in H. Hence, when adding a shortest-path, we only "pay" for the number of missing edges (those that are incident to vertices with degree in $[\Delta_1, \Delta_2]$). We will take care of these paths in $O(\log n)$ phases. For every $i \in \{0, 1, \ldots, 2\log n\}$, define the set Q_i by randomly including each vertex v into Q_i with probability $p_i = C \log n/(\Delta_1 \cdot 2^i)$ for sufficiently large constant C. For every vertex u that has a neighbor, say w, in Q_0, add one edge between u and w to the spanner.

In each phase $i \geq 1$, we take care of all paths $\pi(u, v)$ that have $x \in [2^{i-1}, 2^i]$ edges that are missing in H. This is done as follows. For each $t_1 \in Q_0$ and each $t_2 \in Q_i$, add to H the shortest t_1-t_2 path in G that has at most 2^i missing edges in H. That is, among all paths between t_1 and t_2 in G that have at most 2^i edges that are not in H, pick the shortest one and add its edges to H.

(c1) Prove that H has $\widetilde{O}(n^{4/3})$ edges and **(c2)** show that H is a +6-spanner.

Distance Oracles

This section is devoted for the Thorup and Zwick distance oracles that we have studied in depth in class. We already saw that these oracles are quite "flexible" and can be extended to related settings of labeling and routing schemes. In this exercise, we will reveal another useful extension of these oracles.

Exercise 2. The distance oracle presented in class supported distance queries u, v for any $u, v \in V \times V$. Our goal is to construct a *smaller* oracle of size $o(n^{1+1/k})$ that would handle only a subset of query pairs. In particular, you are now given a subset of vertices, called hereafter, *sources*, $S \subseteq V$ and it is required to design a *source-wise* approximate distance oracle scheme. In such a scheme, the query algorithm receives only queries of the form $u, v \in S \times V$ (i.e., your oracle should only answer distance queries between one of the sources and some other vertex in the graph). Show that in such a case, the preprocessing algorithm can be adapted to yield a data structure of size $\widetilde{O}(k|S|^{1/k} \cdot n)$. Your answer should include a modified preprocessing algorithm, query algorithm and a correctness analysis. Hint: Modify the definition of A_0 and the sampling probabilities.

Routing Schemes

Exercise 3. Describe an efficient routing scheme for the unweighted $\sqrt{n} \times \sqrt{n}$ 2-dimensional grid. The labels and the routing tables should be of size $O(\log n)$ bits. Bonus: extend it to the d-dimensional n-vertex hypercube for $n = 2^d$.